CS 5540 Pattern Recognition

- Prerequisite
  - linear algebra, probability theory, signal processing
- Textbook & references
  - E. Alpaydin, Introduction to Machine Learning, 4th Ed., The MIT Press 2020
  - S. Theodoridis and K. Koutroumbas, Pattern Recognition, 4th Edition, Academic Press, 2009
  - C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
  - R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd Edition, John Wiley, 2001
- Grading
  - Homework 30-35%
  - Computer Experiments (Theodoridis et al) 30-35%
  - Term Project 30-35%
    - Proposal / Oral presentation & demo / Report

Introduction

- Pattern recognition
  - To develop methods that can
    - Classify unknown objects into a number of categories or classes
    - Automatically detect patterns or structure in data
    - Make data-driven decisions

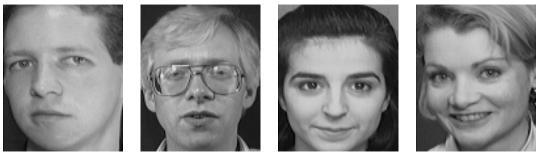
Many data are not completely random, they have structure

  - Is closely related to machine learning and data mining
  - But differs in terms of its emphasis and terminology
    - Machine learning
      - To program computers to optimize a performance criterion using example data or past experience
    - Data mining
      - Application of machine learning methods to large databases
        - » To construct a simple model from big data

Applications (1)

- Machine vision system
  - Automatic visual inspection needs to classify captured objects into the defect or non-defect classes
  - In an assembly line, different objects must be recognized so that a robot arm can place the objects in the right place
  - Face detection and recognition

Training examples of a person



Test images

- Optical character recognition (OCR) system
  - To classify each character into the letter/number/punctuation classes

Applications (2)

- Speech recognition
  - Spoken words recognition
  - Speaker identification
- Document classification
- Email spam filtering

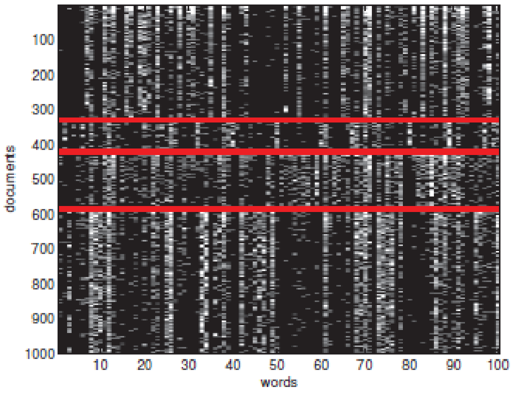


Fig. 1.2 [Murphy 2012]

Figure 1.2 Subset of size 16242 x 100 of the 20-newsgroups data. We only show 1000 rows, for clarity. Each row is a document (represented as a bag-of-words bit vector), each column is a word. The red lines separate the 4 classes, which are (in descending order) comp, rec, sci, talk (these are the titles of USENET groups). We can see that there are subsets of words whose presence or absence is indicative of the class. The data is available from `http://cs.nyu.edu/~roweis/data.html`. Figure generated by `newsgroupsVisualize`.

Why “Learn”?

- Learning is used when [Alpaydin, 2020]
  - Human expertise does not exist
    - Navigating on Mars
  - Humans are unable to explain their expertise
    - Speech recognition, face recognition
      - We do it unconsciously but are unable to explain how we do it
  - Solution changes in time
    - Routing on a computer network
  - Solution needs to be adapted to particular cases
    - User biometrics

Examples (1)

- The fish classifier [Ch1, Duda 01]

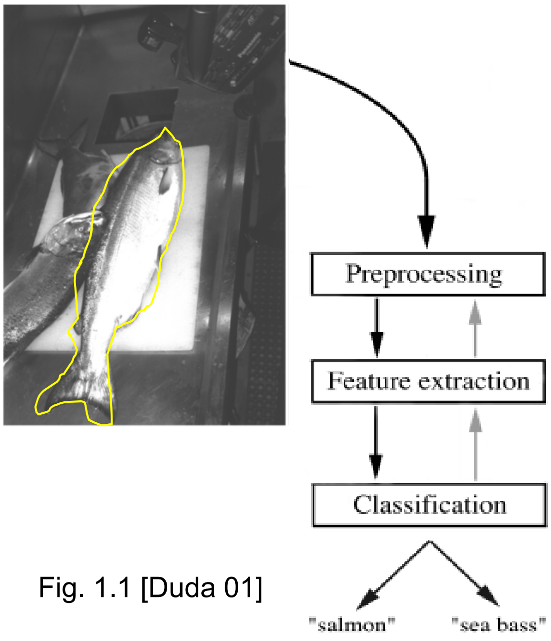


Fig. 1.1 [Duda 01]

- Preprocssing
  - *Segmentation* of the object from background
- Feature extraction
  - The *measurable quantities* from patterns
    - Making the 2 classes distinct from each other
    - Be insensitive to noises (measurement noise, segmentation error)
- Modeling
  - *Description* of each class in mathematical form
- Classification
  - The classifier divides the feature space into class regions

Examples (2)

- The fish classifier (cont.)
  - Input $\mathbf{x}$
    - Descriptive features of fishes
  - Output y
    - Sea bass or salmon
  - Unknown target function $f: X \rightarrow Y$ (or target distribution $P(y|\mathbf{x})$)
    - Ideal formula for fish classification
  - Training set $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$
    - $y_i = f(\mathbf{x}_i), i = 1, \dots, N$
  - Goal
    - To learn a formula $g: X \rightarrow Y$ to make predictions on new inputs
      - g should approximate f on D
      - g is chosen from the hypothesis set H (set of candidate functions): $g \in H$
        - » e.g., set of all linear functions

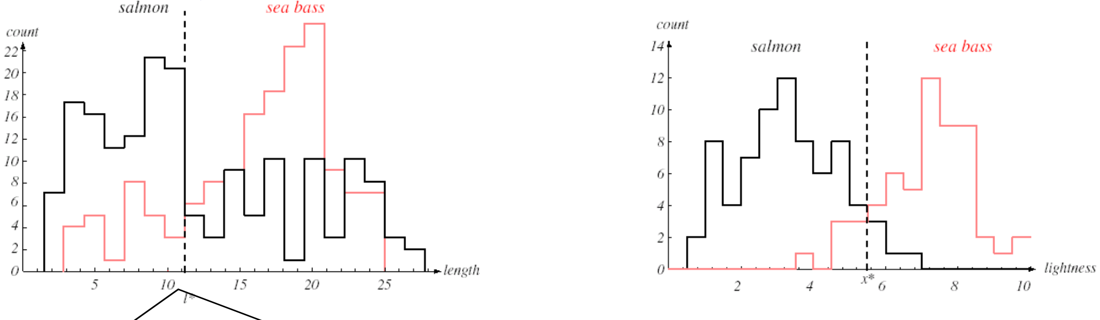
Examples (3)

- The fish classifier (cont.)
  - Training set D
    - Single feature x (treated as a random variable)

Feature x : length

Decision boundary

Feature x : lightness



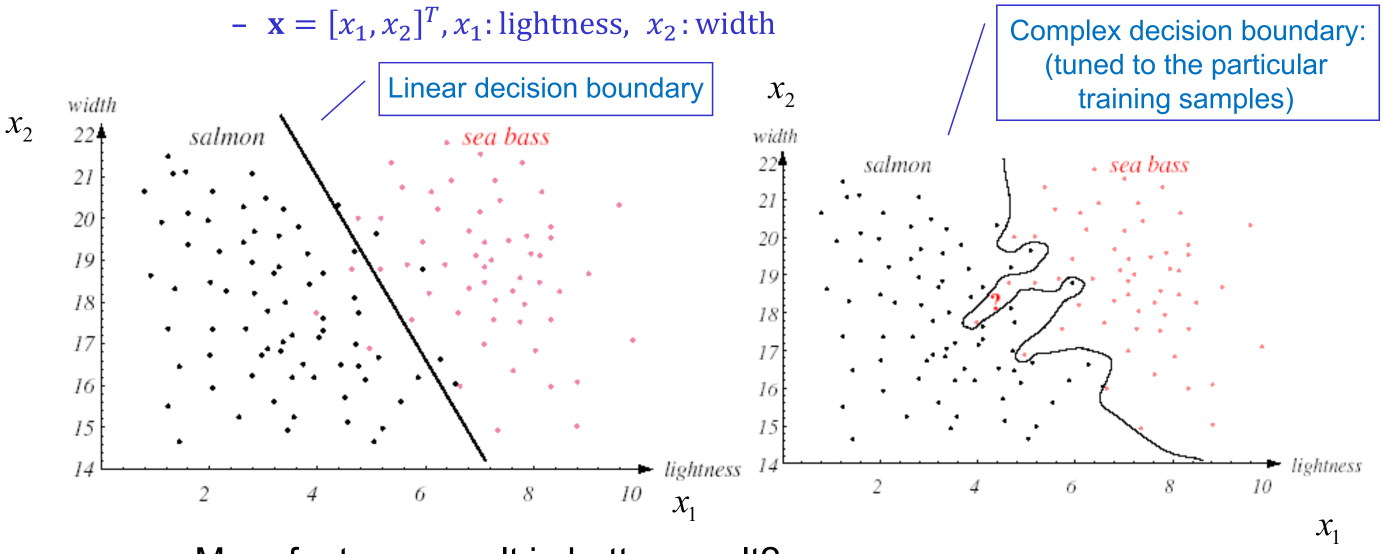
This single criterion is poor on classification

Figs. 1.2 & 1.3 [Duda 01]

- To make a decision rule to minimize the cost

Examples (4)

- Example – the fish classifier (cont.)
  - Training set
    - Feature vector (treated as a random vector)
      - $\mathbf{x} = [x_1, x_2]^T, x_1$: lightness, x_2: width



Linear decision boundary

Complex decision boundary: (tuned to the particular training samples)

- More features result in better result?

Figs. 1.4 & 1.5 [Duda 01]

Examples (5)

- Example – the fish classifier (cont.)
  - Good generalization
    - The classifier can generalize to new patterns that are not part of the training set
      - Even if we get slightly poorer performance on the training samples

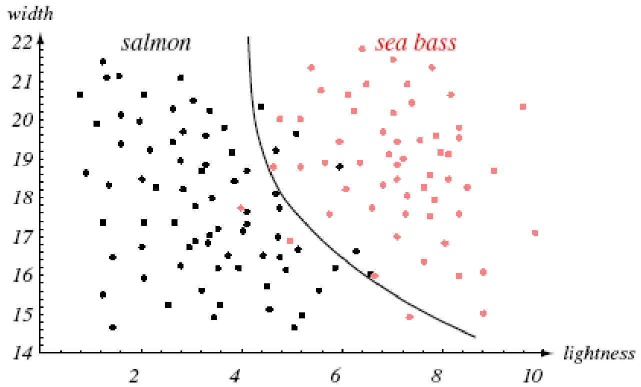


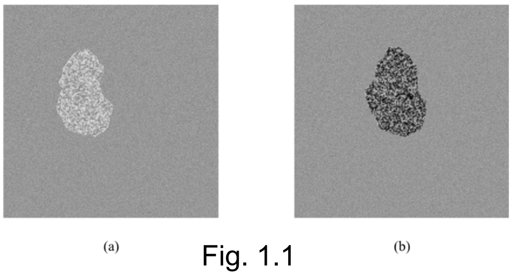
FIGURE 1.6. The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Fig. 1.6 [Duda 01]

Examples (6)

- Medical image classification
  - 2 classes

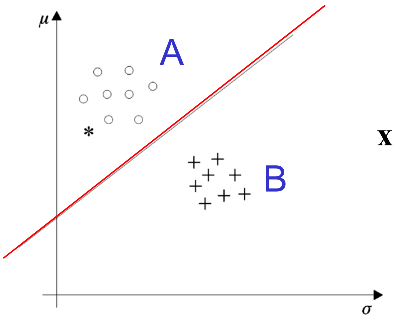
A: benign lesion



B: cancer

Fig. 1.1

  - Features
    - Mean of the intensity
    - Standard deviation around the mean
  - Classifier design
    - The decision line
      - Learned from the training data
  - The unknown pattern $*$ is more likely to belong to class A than class B



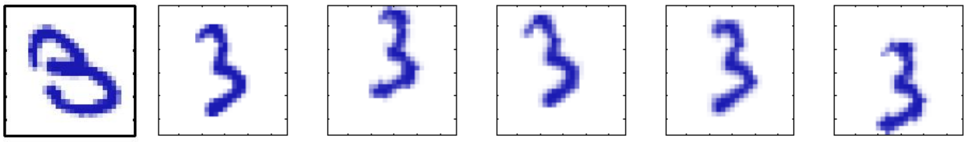
$$\mathbf{x}_i = \begin{bmatrix} \sigma_i \\ \mu_i \end{bmatrix}$$

Fig. 1.2: training samples [Theodoridis 09]

Supervised & Unsupervised (1)

- Supervised learning - classification
  - A set of training data with known class labels is available
    - The output is a finite number of categories $y \in \{1, 2, \dots, M\}$
  - The classifiers are designed by exploiting the a priori known information

  - e.g. hand-written digit recognition
    - Ten classes: 0, 1, 2, …, 9
    - Training data D = { (image, digit), … }



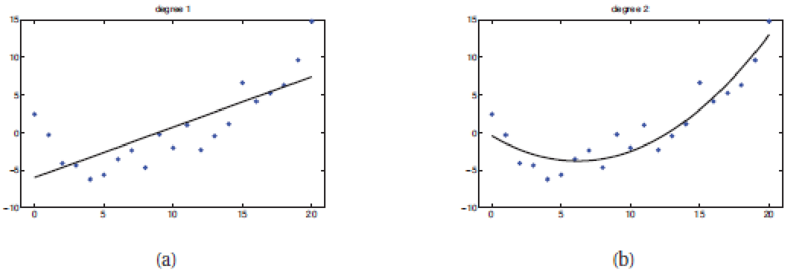
  - The training samples of class “3”

Fig. 12.1 [Bishop 06]

Supervised & Unsupervised (2)

- Supervised learning – regression
  - $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$
    - The output y is continuous (real-valued)
    - $y = g(\mathbf{x}|\theta)$, $g(.)$: the model, θ: parameters
  - Examples
    - Prediction of stock market price
    - Facial age estimation
    - Temperature prediction

$y = wx + w_0$



$y = w_2x^2 + w_1x + w_0$

Fig. 1.7 [Murphy 2012]

Figure 1.7 (a) Linear regression on some 1d data. (b) Same data with polynomial regression (degree 2).

Supervised & Unsupervised (3)

- Unsupervised learning
  - Labeled training data are not available
    - We are only given the data $D = \{\mathbf{x}_i, i = 1,2, \dots, N\}$
  - To gain some understandings of the process that generated the data
  - Clustering
    - To group data into a number of clusters
      - In terms of the similarity between feature vectors and the cluster criterion
    - e.g. coin classification
      - Correct clustering?
      - Number of clusters?

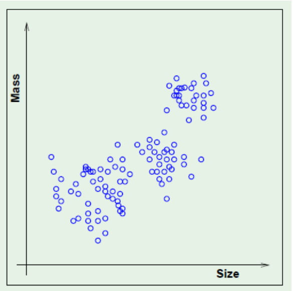


Fig. 1.6(a) [Abs-Mostafa et al., 2012]

Supervised & Unsupervised (4)

- Unsupervised learning (cont.)
  - Discovering latent factors
    - To find a small number of underlying latent factors which capture the essence of the data
    - e.g., facial appearance modeling
      - A few latent factors (such as lighting, pose, identity, or expression) which may describe most of the variability of the high dimensional data

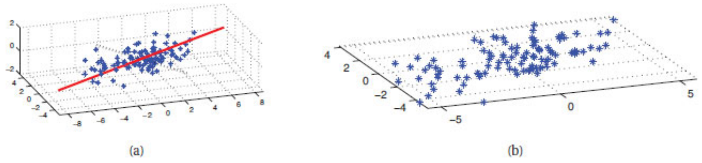


Figure 1.9 (a) A set of points that live on a 2d linear subspace embedded in 3d. The solid red line is the first principal component direction. The dotted black line is the second PC direction. (b) 2D representation of the data. Figure generated by pcaDemo3d.

Fig. 1.9 [Murphy 2012]

  - Matrix completion
    - To infer plausible values for missing data
      - e.g., image inpainting

Supervised & Unsupervised (5)

- Semi-supervised learning
  - A small number of labeled data is available
  - A constrained cluster task

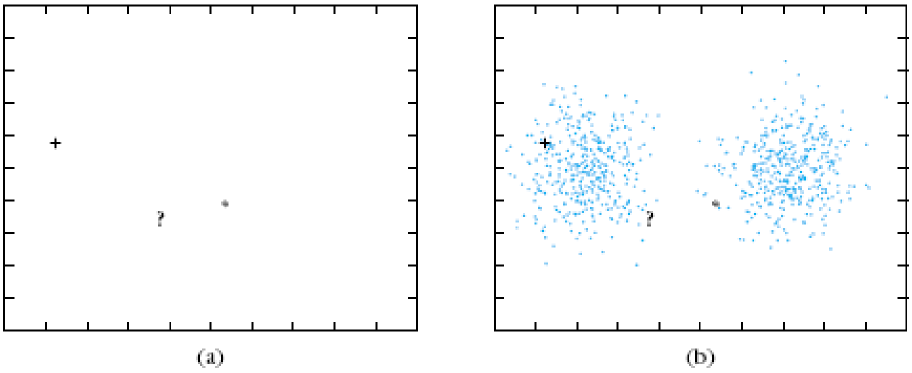
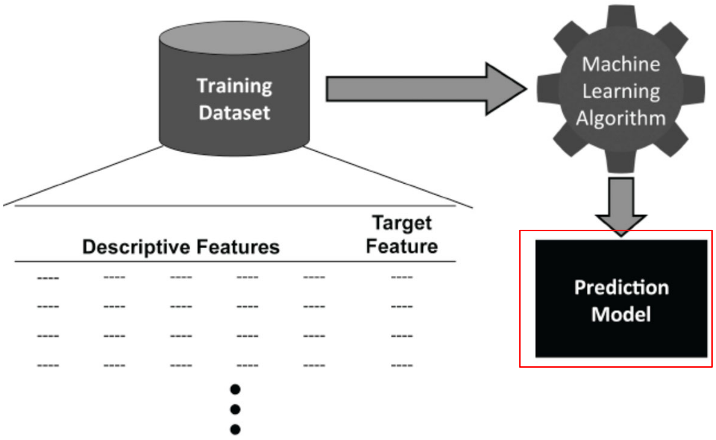


FIGURE 10.5

(a) The unknown point, denoted by "?", is classified in the same class as point "*". (b) The setup after a number of unlabeled data have been provided, which leads us to reconsider our previous classification decision. [Theodoridis 09]

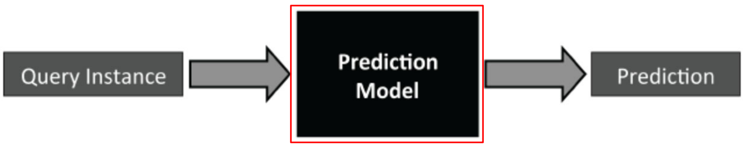
The Design Cycle (1)

- Supervised learning
  - Training stage



  - Testing stage

[Kelleher et al, 2015]

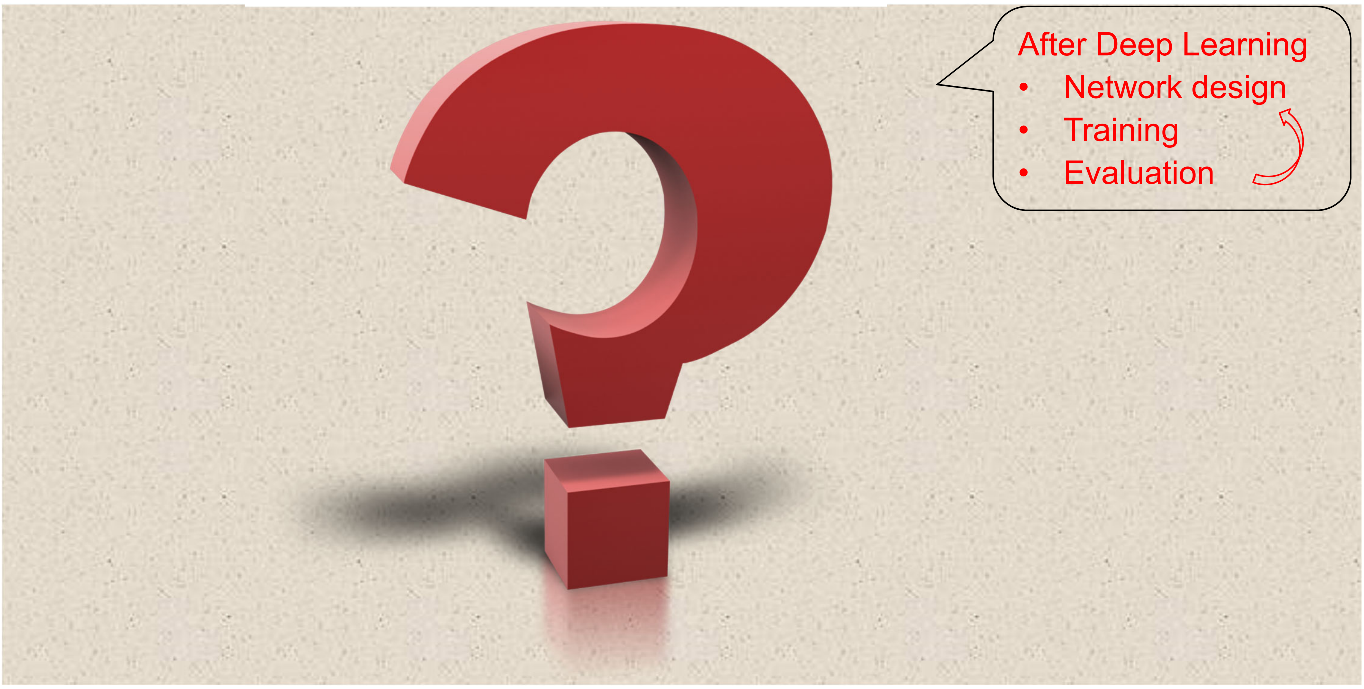


The Design Cycle (2)

- Data collection
- Feature generation
  - Domain-dependent
    - Useful for discriminating classes of interest
    - Account for intra-class variations
  - Robust features
    - Invariant to irrelevant transforms (e.g. translation, rotation, scale)
    - Insensitive to noise, occlusion, …
- Feature selection
  - To find the features that most contribute to the classification
  - Curse of dimensionality
    - Error rate may increase with the number of features when the training set is small

The Design Cycle (3)

- Data collection



After Deep Learning

- Network design
- Training
- Evaluation

The Design Cycle (4)

- Design of predictive model
  - Model choice
    - Searching through a set of possible models
  - Training
    - Ill-posed problem
      - The training set is only a small sample in the domain
      - No unique solution can be determined using only the available information
    - Inductive bias
      - Assumptions that define the model selection criteria
- Performance evaluation
  - Error rate, risk, computational complexity
  - Generalization
    - The capability to operate satisfactorily with data outside the training dataset

The Design Cycle (5)

- Example [Kelleher et al, 2015]
  - To classify customer households into: single, couple, or family
    - Descriptive features of 5 customers
      - Binary attributes of shopping habits
  - The target function
    - $f: X \rightarrow Y$
    - Domain: $2^3 = 8$ combinations of features
    - Range: 3 possible outputs for each instance
    - There are 3^8=6561 possible models could be used

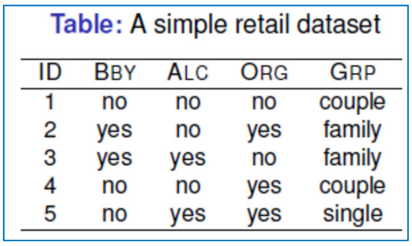
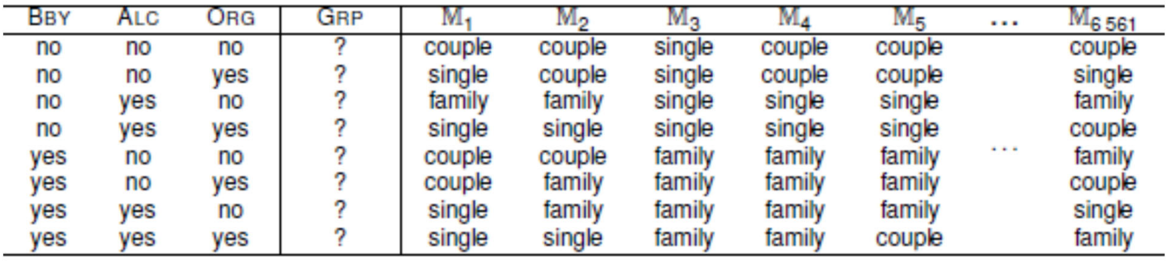


Table: A simple retail dataset

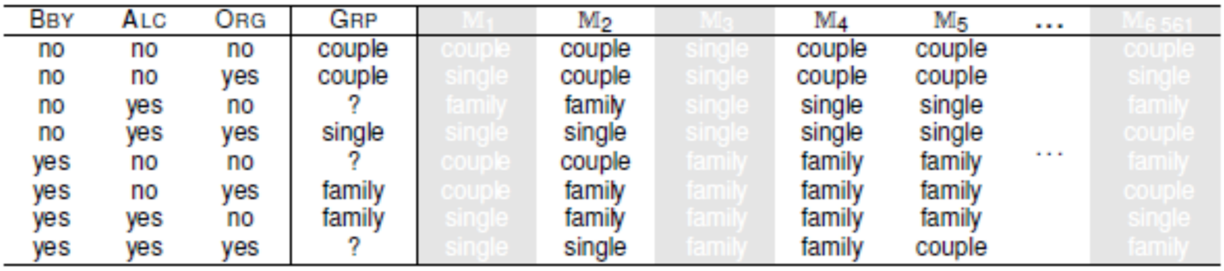
ID	BBY	ALC	ORG	GRP
1	no	no	no	couple
2	yes	no	yes	family
3	yes	yes	no	family
4	no	no	yes	couple
5	no	yes	yes	single



BBY	ALC	ORG	GRP	M_1	M_2	M_3	M_4	M_5	...	$M_{6\,561}$
no	no	no	?	couple	couple	single	couple	couple		couple
no	no	yes	?	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	?	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	?	couple	family	family	family	family		couple
yes	yes	no	?	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

The Design Cycle (6)

- Example (cont.)
  - The set of models that are consistent with the training data



BBY	ALC	ORG	GRP	M_1	M_2	M_3	M_4	M_5	...	$M_{6\,561}$
no	no	no	couple	couple	couple	single	couple	couple		couple
no	no	yes	couple	single	couple	single	couple	couple		single
no	yes	no	?	family	family	single	single	single		family
no	yes	yes	single	single	single	single	single	single		couple
yes	no	no	?	couple	couple	family	family	family	...	family
yes	no	yes	family	couple	family	family	family	family		couple
yes	yes	no	family	single	family	family	family	family		single
yes	yes	yes	?	single	single	family	family	couple		family

- 3^3=27 potential models that are consistent with the training data
  - No unique model can be found based on the training dataset alone
    - » Ill-posed problem
    - » These models may disagree on predictions of query instances
- A consistent model ≈ memorizing the dataset
  - No learning
  - Sensitive to noises in the dataset

What Can Go Wrong? (1)

- Supervised learning algorithms
  - Work by finding the prediction models guided by
    - The training dataset
    - The inductive bias
      - Restrictive bias
        - » The models we consider during learning
          - » e.g., linear model
      - Preference bias
        - » The models we prefer over others
          - » e.g., order of a model
- Under inappropriate inductive bias
  - Underfitting
    - Model is too simple to represent the relationship in the dataset
  - Overfitting
    - Model is too complex

What Can Go Wrong? (2)

- Example [Kelleher et al, 2015]
  - To predict a person's income based on a single feature (Age)
    - The dataset

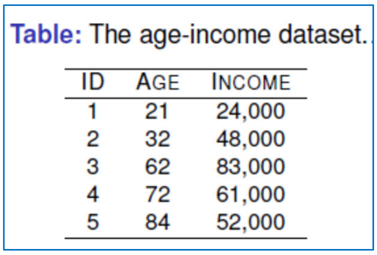
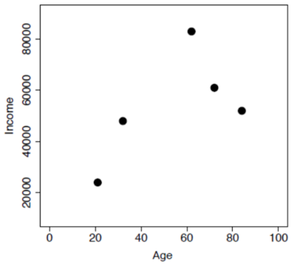
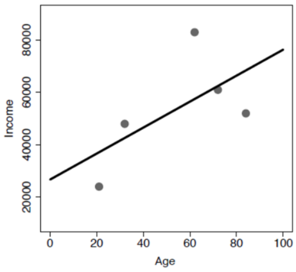


Table: The age-income dataset.

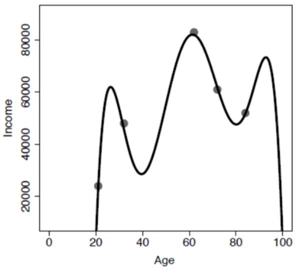
ID	AGE	INCOME
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000



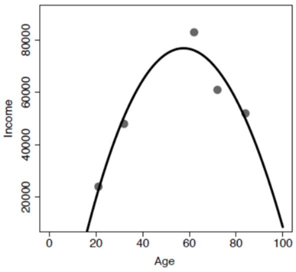
- The models
  - underfitting



overfitting



good generalization



What Can Go Wrong? (3)

- Example [Duda, 01]
  - Overfitting
    - A 10th-degree polynomial fits the training data with zero error
      - A small change in the data will change the parameters significantly
      - The generalization error is much higher for this fitted curve

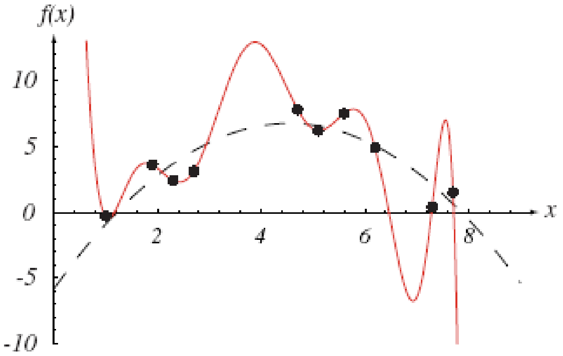


Fig. 3.4 [Duda 01]

FIGURE 3.4. The "training data" (black dots) were selected from a quadratic function plus Gaussian noise, i.e., $f(x) = ax^2+bx+c+\epsilon$ where $p(\epsilon) \sim N(0, \sigma^2)$. The 10th-degree polynomial shown fits the data perfectly, but we desire instead the second-order function $f(x)$, because it would lead to better predictions for new samples. From: Richard

What Can Go Wrong? (4)

- Another example

Muller et. al, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 181-201, 2001.

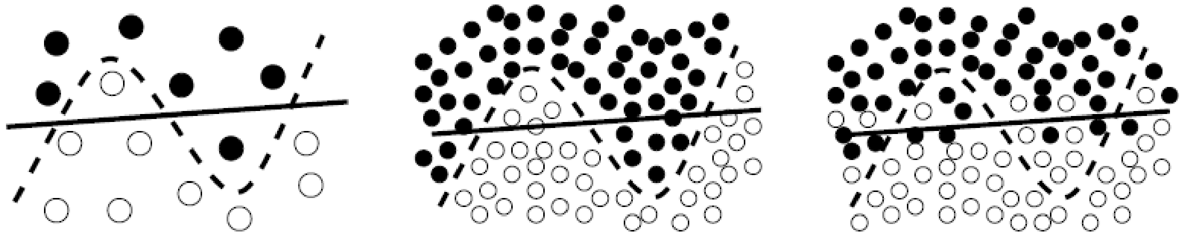


Fig. 1. Illustration of the overfitting dilemma: Given only a small sample (left) either, the solid or the dashed hypothesis might be true, the dashed one being more complex, but also having a smaller training error. Only with a large sample we are able to see which decision reflects the true distribution more closely. If the dashed hypothesis is correct the solid would underfit (middle); if the solid were correct the dashed hypothesis would overfit (right).

Summary

- To develop (data-driven) methods that can
  - Automatically learn the relationship between
    - A set of descriptive features, and
    - The output labels
  - Automatically detect latent structure in data

- Ill-posed problem
  - Generalization performance
  - Inductive bias
  - Underfitting
  - Overfitting

Course Contents

- Supervised learning
- Bayesian Decision Theory
- Parametric Methods
- Multivariate Methods
- Dimensionality Reduction
- Clustering Nonparametric Methods
- Linear Discrimination
- Multilayer Perceptrons
- Kernel Machines
- Combining Multiple Learners